Video Scene Categorization by 3D Hierarchical Histogram Matching

Paritosh Gupta[1], Sai Sankalp Arrabolu[1], Mathew Brown[2] and Silvio Savarese[1]
[1] University of Michigan, Ann Arbor, USA [2] University of British Columbia, Vancouver, Canada
{paritosg, saisank, silvio}@umich.edu mbrown@cs.ubc.ca

Abstract

In this paper we present a new method for categorizing video sequences capturing different scene classes. This can be seen as a generalization of previous work on scene classification from single images. A scene is represented by a collection of 3D points with an appearance based codeword attached to each point. The cloud of points is recovered by using a robust SFM algorithm applied on the video sequence. A hierarchical structure of histograms located at different locations and at different scales is used to capture the typical spatial distribution of 3D points and codewords in the working volume. The scene is classified by SVM equipped with a histogram matching kernel, similar to [21, 10, 16]. Results on a challenging dataset of 5 scene categories show competitive classification accuracy and superior performance with respect to a state-of-the-art 2D pyramid matching methods [16] applied to individual image frames.

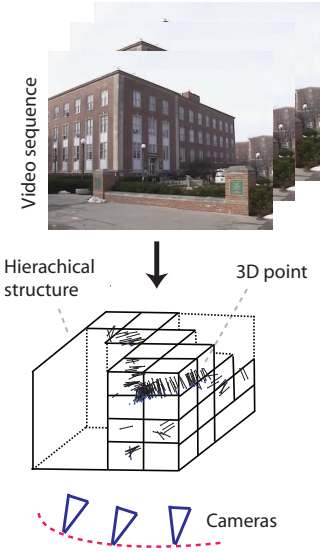

Figure 1. The basic scheme.

1. Introduction

Cheap and high resolution sensors, low cost memory and increasing bandwidth capacity are enabling individuals to capture and manipulate visual data more easily than ever. Current technology allows users to point their cellphone at a scene, acquiring low resolution video sequences that capture relevant visual information, and send that data to a friend somewhere else in the world. It is desirable to go beyond this and further process the acquired imagery for extracting useful semantics. Users would benefit from having an algorithm that is able to answer basic questions such as: what am I looking at? what are the objects in the scene? Among these, it is crucial to enable the interpretation of the overall semantic of the scene, and thus, the recognition of the category the scene belongs to. Is this an outdoor or indoor scene? A park, a neighborhood in suburbia or the parking lot of a shopping mall? This would allow the identification of the context where the action takes place and help extracting the semantic of specific objects (such as, cars, trees, buildings) with higher degree of accuracy and lower false alarm rates. This capability is also useful in a number of applications such as automatic annotation of *street view* imagery [1] and autonomous navigation. Recognizing scene categories from medium-low resolution video sequences (that is, video sequences acquired from inexpensive consumer hand-held cameras or cell phone devices) is the focus of this paper. A critical issue that we address in this work is the ability to design algorithms that are robust and efficient, and thus useful in a real time settings.

The problem of recognizing scene categories from single 2D images has received increasing attention during the past few years. Researchers have proposed a wide range of different representations: from holistic descriptions of the scene [22] to interpretation of the scene as collection of features or intermediate topics, [8, 29, 4], with more or less [8, 25] degree of supervision during the learning process. In these models, the scene is represented as collections of features where the spatial coherency is not preserved. Recent works by [10, 16] have shown that it is possible to incorporate spatial information for efficiently recognizing large number of scene categories. Here, the typical 2D layout of appearance elements across instances is

learnt as part of an underlying 2D pyramid structure. Critically, these methods propose to encode the spatial information in terms of 2D spatial locations only, while no additional 2D/3D geometrical concepts are considered. Recent works have proposed ideas for extracting geometrical properties of the scene, such as vertical/horizontal geometrical attributes [12], approximate depth information [24], as well as using semantic [28] or geometrical context for improving object detection [13, 5, 7]. However, none of these methods have used explicit 3D geometrical reasoning for classifying scene categories.

We argue that using the underlying 3D structure of the scene can greatly help toward the goal of scene categorization. We propose to extract this information from video sequences where the same scene is observed for a short amount of time by a moving camera. Since we would like to work with medium or low definition video sequences (where no information about the camera parameters is in general available), robust techniques for extracting and interpreting 3D information must be used. We propose to employ recent structure from motion algorithms [6] (Sec. 2) for solving the full un-calibrated SFM problem. The result is still a fairly sparse reconstruction of 3D points and camera locations. This makes most of state-of-the-art methods for 3D shapes classification [23, 14, 11, 9, 15, 26, 17] inadequate. In these methods the underlying reconstructed structure is assumed to be dense and accurate, and appearance information is most of times ignored.

Thus, our challenge is to find a representation that can be built from highly sparse reconstructions and low resolution imagery but at the same time is able to capture the geometrical and appearance essence of a scene category. We propose to represent a scene by looking at the typical distributions of 3D points along with appearance information for characterizing a generic urban scene category. In our model, each 3D point is labeled using a dictionary of codewords capturing epitomic appearance elements of the scene imagery. Then, a collection of histograms of codewords computed at different locations and scales within the working space is used to model the scene. Such collection is organized in a 3D hierarchical structure as explained in Sec. 2 and is recursively built based on the statistics of occupancy of points in the 3D space across all the categories. Unlike previous work on scene categorization, our model is robust with respect view point variability as discussed in 2.3. Finally, video sequences are categorized with a non linear SVM classifier using a matching kernel similar to the one proposed by [21, 10, 16] (Sec. 3). A number of experiments with a 5-class scene dataset of low resolution video sequences demonstrates that the added 3D spatial information is indeed critical for obtaining more accurate scene classification (Sec. 4).

2. Scene representation

2.1. Overview

Our goal is to learn models of scene categories from single video sequences and use these models to categorize query video sequences. In this section we explain in details our proposed representation for modeling a scene from video sequences. Let us denote by c a scene category and by s a video shot capturing a specific scene of category c. The first step is to recover the scene structure (3d points) and camera location from the video sequence s. This can be implemented by using state of the art SFM techniques as explained in Sec. 2.2. The reconstructed 3D points along with the camera locations are used to fix a local reference system and a working volume V^o (Fig. 1). The working volume is defined as the 3D volume that encloses the majority of reconstructed 3D points associated to s (Sec. 2.3). This steps is critical if one wants to guarantee that a scene structure has consistent alignment and scale across different instances $s_1, s_2...s_n$ of the same scene class.

The next step is to transfer appearance information from the images (frames) of the video sequence to each reconstructed 3D point. This can be easily done since 3D points are associated to matched feature key points across the frames of the video sequence s_i, as explained in (Sec. 2.2). Appearance information is encoded by labeling each image key point using a dictionary of learnt codewords. Image key point labels are transferred to the corresponding 3D point using a voting scheme (Sec. 2.4).

Once each 3D point is associated to a codeword label, the spatial distribution of such codewords in the working volume must be captured. Inspired by some of the previous works in 3D shape matching [11], we model such distribution by using histograms. In our work each histogram is capturing the frequency of occurrences of codewords in a sub volume V^l: The ensemble of such histograms computed at different sub-volume locations and dimensions are used to model the overall distribution of codewords in V^o. In practice, a hierarchical structure of sub-volumes is constructed by recursively subdividing the portion of V^o into smaller sub-volumes V^l (Sec. 2.5).

We claim that the 3D hierarchical structure of histogram of codewords is a good representation for modeling the interclass and intra-class scene variability (different scene categories differ in terms of their overall codeword label distribution as well as their multi-scale spatial distribution in the 3D working volume). Furthermore, we claim that generalization within each scene category is achieved because: i) scene shape variability across instances of the same scene category is accommodated by the "bag-of-words" paradigm built on top of multi-scale hierarchical structure; ii) appearance variability is accommodated by introducing the vocabulary of codewords.

Critically, a hierarchical pyramid structure for histograms of codewords has been proposed for modeling scene categories in 2D images [16] and has been proven to produce high classification rates. Our method, however, is not just an extension of [16] to 3D but it differs in one important aspect. The spatial pyramid structure in [16] recursively decomposes the image into quadrants following 2^{2l} progression. Each stage of the decomposition l is called *level*. The natural extension of the spatial pyramid to 3D would be to recursively decomposing the working volume into eight equal cubic octants following a 2^{3l} progression; thus at level l the 3D decomposition has 2^l times more bins. Notice, however, that, unlike the 2D case where features statistically occupy the image in an almost uniform fashion across categories, in the 3D case points tend to conglomerate into specific regions in the working volume - that is, points occupy sparse locations in the 3D space (Fig. 5). The consequence of this is clear: as the level of decomposition increases, the percentage of empty octants quickly increases, leaving only a sparse and limited number of octants embedding the actual scene structure. Thus, rather than subdividing the whole volume using a *blind* pyramid decomposition scheme, we only decompose volumes that are *likely* to contain scene structure. We call this scheme an *occupancy* decomposition scheme (Sec. 2.5).

2.2. Structure from Motion

The first step of our algorithm is to generate the 3D geometry of scene and camera locations from our input video sequences. We use a Structure and Motion solver similar to [6]. This begins by extracting SIFT [18] key-points from the input video sequence, resampled at 1 frame / second. Consistent 2-view matches are found via robust solution for the Fundamental Matrix using RANSAC. Initial images for bundle adjustment are selected using a 3D information criterion similar to GRIC [27]. From here, bundle adjustment proceeds in a metric coordinate frame. Each camera is parameterized by a rotation matrix, translation and focal length, and these values are initialized by copying the parameters of the best matching image. Images are added one by one, with a pose estimation step with fixed structure preceding joint optimization over all cameras and structure. The output of this step is a cloud of 3D points and the location and pose of the cameras. Fig. 2 shows a few examples of reconstructed geometry. Notice that we do not need to use any prior knowledge about the camera pose or scene geometry to obtain such reconstruction. As a result of the reconstruction, 3D points are set in correspondence to image key points, and image key points are linked across the 2 or more frames of the video-sequences if they all correspond to the same 3D point (tracks) (Fig. 4). Experimental validation shows our average re-projection error is less than one pixel.

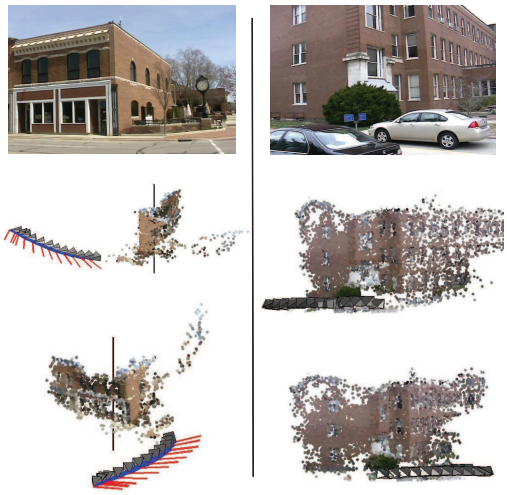

Figure 2. Examples of 3D reconstructions.

2.3. Aligning the Working Volume

The reconstructed 3D points along with the camera locations are used to locate, re-scale and orient the working volume V^o in the world reference system. This step is critical in order to guarantee that a scene structure has consistent alignment across different instances $s_1, s_2 ... s_n$ of the same scene class, thus making the 3D representation scale, rotational and translational invariant. The working volume V^o is defined as a cube of side d that encompasses the majority of 3D points. We set $d = 2\sigma$, where σ is the standard deviation of the distribution of 3D points in space and normalize (rescale) the cube size so as to have a cube side of unitary length. The orientation of V^o in space requires more careful analysis. It is clear that V^o can be locked in 3D if the orientation and direction of two (normal) vectors are determined. One normal direction and orientation is locked by estimating the normal of the ground plane.

We estimate the ground plane using a source of metadata that the camera-person unconsciously provides via the camera trajectory. To do this, we make use of the following assumptions: 1) The camera is kept at a constant height; 2) The user does not twist the camera relative to the horizon; 3) The ground plane is flat (i.e. the plane normal is aligned with gravity). In practice, assumptions 1 and 2 are obeyed quite well by even an amateur camera-person, and assumption 3 is also reasonable for our sequences. Given that these assumptions hold, the camera x-axes and centres of projection all lie in the same plane (the ground plane). We can combine these sources of information by finding the normal to the plane containing the camera motion vectors and x-axis directions

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \mathbf{u}^T \mathbf{C} \mathbf{u} \ , \tag{1}$$

where $\mathbf{u}$ is a unit vector and $\mathbf{C}$ is given by

$$\mathbf{C} = \sum_i \mathbf{u}_x^{(i)} \mathbf{u}_x^{(i)T} + \sum_i \mathbf{u}_m^{(i)} \mathbf{u}_m^{(i)T} \ . \qquad (2)$$

$\mathbf{u}_x^{(i)}$ is a unit vector parallel to the x-axis of the ith camera, and $\mathbf{u}_m^{(i)}$ is a unit motion vector between that camera and another camera selected at random from the sequence. This gives equal weight to the information provided by assumptions 1 and 2. Note that there is a degeneracy in this procedure if the motion vectors and camera x-vectors are all parallel, in which case there is a 1 parameter family of valid normal vectors. However, this is unlikely to occur in practice as it would require the camera to translate exactly sideways along its x-axis in all frames.

A second normal can be estimated by assuming that (at least) one dominant planar surface exists in the scene. This is a reasonable assumption as we are focussing on classifying urban scene categories that are likely to contain vertical planes such as walls, fences, or facades. The orientation of the cube can be fixed using this second normal.Such planar surfaces can be identified by analyzing the distribution of normal vectors computed from the 3D points (Fig. 3). Standard techniques can be used for robustly estimating the normals from a neighbor of 3D points. Normals can be used to build a co-variance matrix whose eigenvalues indicate the modes of the distribution. The first mode corresponds to the first dominant plane. The remaining ambiguity - the cube orientation is defined up a 180 rotation - can be resolved by using the visibly constraint: the normal vectors must be pointing toward camera view centers (Fig. 3). Notice that other methods based on pyramid matching [21, 10, 16] make no attempt to set a reference system in 2D (for achieving rotational or scale registration).

Experimental analysis shows that this registration procedure is very robust for urban scenes. Our quantitative analysis (based on visual inspection) shows that the rough location of the ground plane is correctly estimated about 95% of times and that most of the sequences do contain a dominant plane (thus, a dominant normal orientation). Notice that we obtain successful alignment even when no corners (plane intersections) are detectable in the video sequence. Some examples are reported in Fig. 3.

2.4. Codeword Dictionary and Labeling

Next, appearance information must be transferred from the images (frames) of the video sequence to each reconstructed 3D point. This task is easy since 3D points are associated to matched image key points across the frames of the video sequence (Sec. 2.2, Fig. 4). First, a dictionary of codewords is constructed to capture epitomic 2D local appearance information across instances and category. This is done by clustering descriptors associated to image key points (extracted from training images) and assigning codewords labels to each cluster center. Then, each keypoint

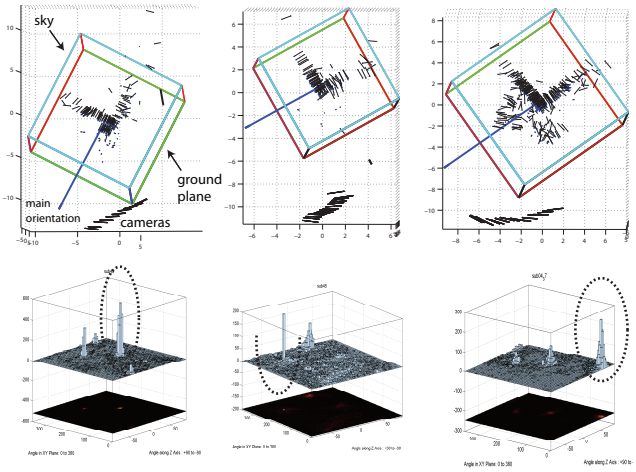

Figure 3. Computing the orientation of the working volume V^o in the world reference system is critical in order to guarantee that a scene structure has consistent alignment across different instances. See text for details. Top row: The reconstructed 3D points along with the camera locations are used to locate and orient the working volume V^o in the world reference system. Green lines indicate the ground plane; cyan lines define the sky plane. The blue normal indicates the plane facing the cameras (viewer). Bottom row: Distribution of normal vectors computed from the 3D points. The main mode of this distribution (highlighted by the circle) corresponds to the dominant plane in the scene.

in each image is assigned to a codeword based on descriptor similarity. Finally, image key point codeword labels are transferred to the corresponding 3D point. Since codewords labels may not be in agreement, a simple voting scheme is used to select the actual 3D point label. Specifically, the label with highest percentage of occurrence among all matched key-points is selected. The percentage of occurrence may be used to prune out 3D points whose label is assigned with low confidence.

2.5. The hierarchical spatial structure

Once each 3D point is associated to a codeword label, the spatial distribution of such codewords must be captured at different scales and different locations in the working volume V^o (hierarchical spatial structure). We will first illustrate the simpler case of modeling such distribution using a 3D pyramid structure H of histograms of codeword labels.

Pyramid decomposition scheme. We proceed by decomposing the working volume V^o into a pyramid structure of sub-volumes. This is similar to an octree subdivision scheme where V^o is partitioned by recursively subdividing it into eight octants $V_1^l ... V_8^l$ (Fig. 1). If we denote by L the last level of subdivision, it is easy to verify that the number D of partitions at level L is $D = 2^{3L}$. The pyramid structure $H(L)$ is obtained as an ensemble of histograms H^l of codewords computed in each sub-volume for each

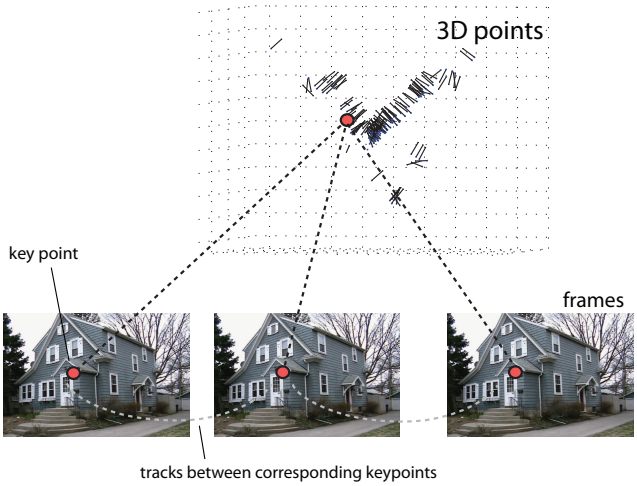

Figure 4. As a result of the reconstruction, 3D points are set in correspondence to image key points, and image key points are linked across the 2 or more frames of the video-sequences if they all correspond to the same 3D point (tracks).

level of subdivision l. H^l is obtained by concatenating 2^{3l} histograms computed for all of the 2^{3l} sub-volumes for level l. Histograms are concatenated so as to be suitable for SVM classication when equipped with a pyramid matching kernel (Sec. 3).

Occupancy-based decomposition scheme. It is clear that as the level of the pyramid structure increases, the histograms are computed on smaller supports, hence increasing the resolution of the overall the representation. As mentioned in Sec. 2.1, one drawback of this decomposition scheme is that, as the level increases, the number of octants that remains empty becomes higher and higher. Using the database introduced in Sec. 4 we have calculated the statistics of occupancy of each octant for each level computed across sequences and across categories. The results are reported in Fig. 5,(a). As the figure shows, at level 0, there is obviously only one volume that contains all the points; similarly, at level 1, all of 8 octants (sub-volumes) are occupied by $3D$ points. However, at level 2 we estimate about 40% of empty octants; this number becomes exponentially smaller as the number of level increases. Even if the number of categories increases we still expect some portions of the cube to be empty. This suggests that a simple pyramid decomposition: i) produces a large number of uninformative octants that yield unnecessary long histograms; ii) as the level increases, the size of each octant quickly reaches small volumes (at level 2, $V_2 = V_0/64$; at level 4, $V_2 = V_0/4096$), whereas a slower decay would be more adequate in capturing the scene structure across scales.

We propose to decompose the working volume as follows. This decomposition is constructed once per all by looking at the statistics of occupancy of 3D point across categories for a validation set. First, the level-zero volume

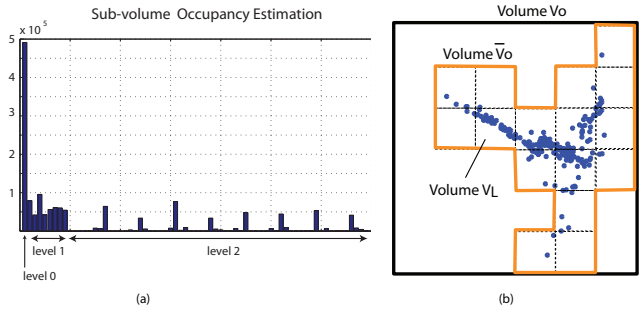

Figure 5. (a) Occupancy (that is, number of 3D points) within each sub-volumes (octants) for different levels for the dataset introduced in Sec. 4. (b) Anecdotal example of distribution of points in a volume V^o. The new working volume $\bar{V}^o$ (outlined in orange) is defined as the collections of level-L octants that have a level of occupancy greater than a threshold T.

V^o is recursively decomposed in octants by following the pyramid decomposition scheme described above until level L. L defines the granularity of our representation. Second, the level zero volume is redefined as $\bar{V}^o$ - that is, as the collection of those level-L octants that contain a number of 3D points greater than a threshold T with probability p (Fig. 5(a)). Thus, octants that tend to be empty most of the times are excluded. T, L and p are determined empirically. Third, $\bar{V}^o$ is recursively randomly decomposed into sub-volumes using a quadratic or linear progression function. The structure of histograms $\bar{H}(L)$ is now obtained as the ensemble of the histograms $\bar{H}^l$ of codewords computed in each sub-volume for each level of subdivision l of $\bar{V}^o$. More specifically: $\bar{H}(L) = \{\bar{H}^o, \bar{H}^1, ...\bar{H}^l, ...\bar{H}^L\}$, where $\bar{H}^l$ is the histogram in $\bar{V}^o$; $\bar{H}^l$ is obtained by concatenating 2^l histograms computed for all of the 2^l sub-volumes for level l. Again, these histograms are matched using a SVM classification machinery (Sec. 3).

Computational efficiency. One clear advantage of the occupancy-based decomposition scheme is that it is computationally more efficient than the basic pyramid one: Fewer and fewer cubes are recursively decomposed at each iteration (level) – that is, only cubes that contain more than T points with probability p are further processed; This results in having a structure $\bar{H}(L)$ of concatenated histograms with a reduced number of bins, and thus, a matching procedure that is faster and more efficient.

View point invariance. We note that this representation for scene categories is robust with respect to view point changes. The reason is three-fold: i) the underlying 3D structure is merely view point invariant thanks to the alignment procedure discussed in Sec. 2.3; ii) each histogram captures a distribution of codewords which are obtained by vector quantizing SIFT descriptors which are known to be robust with respect to small view point changes [19]; iii) the distribution of codewords within each sub-volumes summarizes the appearance of the scene from several vantage

Figure 6. Examples of frames from our dataset of 5 scene categories videos.

points; indeed, codewords are assigned to 3D points which are extracted from tracks of features across frames (Fig. 4); thus, subvolumes include a redundant number of 3D points associated to multiple observations of the same scene from different vantage points; this enables partial view point appearance invariance.

3. Discriminative Model Learning

In Sec. 2 we have proposed a new representation for modeling a scene from a video sequence. Our representation is built on the 3D histogram structure $\bar{H}(L)$ as discussed in Sec. 2.5. From now on, we simplify the notation by suppressing the bar in $\bar{H}$ and $\bar{V}$. By using a suitable kernel, it is possible to learn a SVM classifier for discriminating 3D histogram structures $H(L)$ belonging to different scene classes. The kernel is chosen as the weighted sum of histogram intersections (also called, the *3D matching kernel*), similarly to those originally introduced by [10, 16]:

$$K(H_i(L), H_j(L)) = w_o I(H_i^o, H_j^o) + \sum_{l=1}^{L} w_l I(H_i^l, H_j^l)$$

where the histogram intersection I is defined as

$$I(H_i^l, H_j^l) = \sum_{k=1}^{D} min(H_i^l(k), H_j^l(k))$$

and where L is the level of decomposition, $D = 2^l$ is the total number of cells of a 3D histogram structure of level l; and w is the weight of the level and is calculated as inversely proportional to the volume of the octant at level l. Note that this is a Mercel kernel since it is constructed as a linear combination of histogram intersections I which are shown to satisfy the Mercel condition [21, 10].

4. Experimental Results

We tested the ability of our method to categorize query video sequences. We validate our algorithm with respect to a challenging dataset [2] comprising 5 scene categories: 'downtown', 'suburbia', 'campus', 'shopping mall', 'gas station'. Each category contains 23 short video sequences (400 frames in average). Each video sequence has a resolution of 720×480 pixels per frame. The videos are captured with a consumer portable camera, with unstable camera motion and under very generic poses mimicking an user walking on a sidewalk. Examples of frames from videos in our database are shown in Fig. 6. Even if the scene categories share similar appearance, subtle differences across categories are noticeable. For example the campus tends to have a larger number of windows, the malls tend to show shorter roof structures. In our experiments, only about 5% of some 400 frames per sequence were automatically selected by the SFM algorithm and used for the actual reconstruction. Each frame of each video sequence contained around $2000 - 3000$ SIFT descriptors, whereas the reconstruction (obtained from a given video sequence) contained approximately $10000 - 20000$ 3D points in total. The video sequences were divided in a training and testing set using a leave-one-out (LOO) scheme. This way, at every step of the LOO, as many as 22 video sequences were used in training and one in testing, for a total number of 23 video shots per category being tested. The dictionary of codewords as well as the structure of decomposition of the working volume were learnt separately in order to avoid contamination.

We validated our method using the occupancy-based 3D hierarchical structure discussed in Sec. 2.5. We reported 5-class classification results in Fig. 7. The base volume $\bar{V}^o$ was estimated as 55% of the initial volume V^o. $\bar{V}^o$ was decomposed following a quadratic progression. As the figure shows, this subdivision scheme produces the highest performance (72.2%) at the third level of decomposition (with volume size $= \bar{V}^o/16$). This indicates the optimal level of decomposition of the 3D structure. After that level, performances dwindle down. Notice that the histogram length at level 3 is just 29 bins, which makes the construction of the kernel matrix very efficient. These results were obtained using a dictionary of 200 codewords. Different dictionary sizes produced either inferior or equivalent results.

Furthermore, we have compared our method with the 2D spatial pyramid matching algorithm for 2D scene classification [16]. This experiment is useful for bench-marking our results. The method was applied to individual frames of the video sequence. Since multiple frames are available from the video sequence, and the choice of the frames may affect the classification results, we randomly selected N frames from each video sequence in testing and computed the classification accuracy as the average across the N frames. In our experiment $N = 5$. Fig. 9 shows the average 5-class

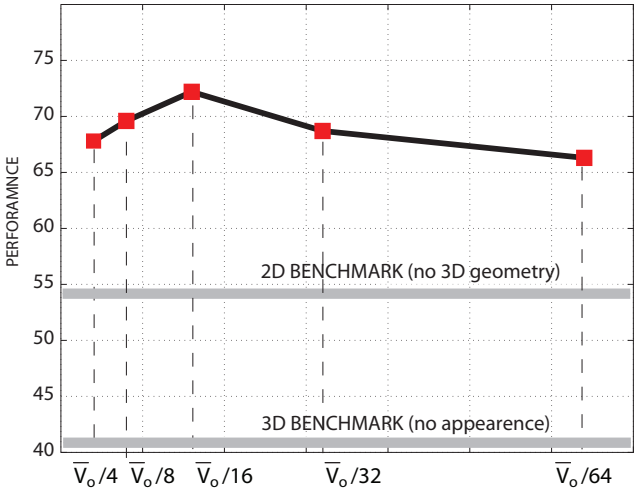

Figure 7. Overall classification accuracy for a 5-class recognition experiment using occupancy-based 3D hierarchical structure. Performances are plotted as function of the level of decomposition of the initial volume $\bar{V}^o$. The best performances (72.2%) are obtained at the third level of decomposition.

classification accuracy for three levels of the pyramid, and for several values of the dictionary size. The corresponding standard deviation is depicted as a vertical bar by each data point. Notice that the best performances (54%, obtained for $L = 2$) are 18.2% lower than the ones observed for the 3D case. Performances for $L > 2$ appear to be lower than 54%. A similar behavior was reported in [16]. Also, notice that performances are overall quite low. This is not surprising given that the scene categories in our dataset are all urban scenes and share very similar appearances. This also suggests that our dataset is a good starting point for validating algorithms for urban scene classification. Classification accuracy for individual classes is reported in the confusion table in Fig. 8.

Finally, we have compared our algorithm with two 3D shape matching methods where the appearance information is partially or fully ignored. The first comparison was done by using the same 3D spatial hierarchical scheme as discussed above. The idea is to eliminate the contribution of appearance information by utilizing dictionaries of codewords of reduced size. When the dictionary size is 1 (i.e., there is only one codeword), no appearance information is encoded. Results are summarized in Table 1. Notice that as the level of decomposition increases the hierarchical structure starts capturing stronger and stronger information about the 3D layout of the scene categories. The best results however (which are achieved for level $L = 4$) are still significantly lower than those obtained using the complete scheme.

The second comparison is made by replacing codewords using vector quantized local shape descriptors, i.e rather than labeling each 3D point with codewords computed by

	CMP	MALL	DWN	GAS	DWN
CMP	65.22		8.70	8.70	17.39
MALL		60.87	13.04	17.39	8.70
DWN	13.04	13.04	52.17	17.39	4.35
GAS		13.04	4.35	73.91	8.70
DWN	34.78	8.70	13.04	21.74	21.74

	CMP	MALL	DWN	GAS	DWN
CMP	78.26	8.70	4.35	4.35	4.35
MALL		69.57	8.70	8.70	13.04
DWN		17.39	65.22	4.35	13.04
GAS	4.35	8.70		78.26	8.70
DWN	8.70	13.04	4.35	4.35	69.57

Figure 8. Left: Confusion table showing classification accuracy using 2D pyramid matching framework (level two; 200 codewords). Right: Confusion table showing classification accuracy using the occupancy-based 3D structure matching framework (level 3; 200 codewords).

	level 0	level 1	level 2	level 3	level 4
NA	0.21	0.27	0.35	0.42	0.43

Table 1. 3D Benchmark comparison table. NA: results using our 3D hierarchical structure with no appearance (dictionary size=1).

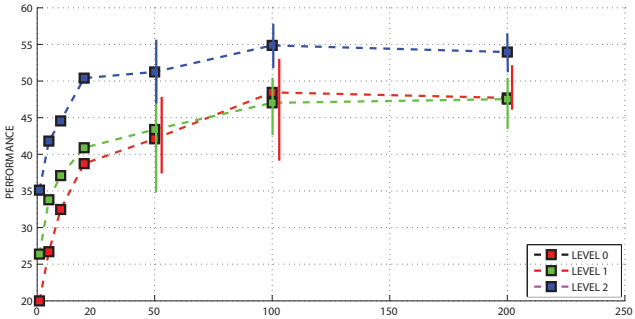

Figure 9. Overall classification accuracy for a 5-class recognition experiment using the 2D spatial pyramid matching algorithm [16]. The figure reports performances for three levels, and several values of the dictionary size. No significant improvement is observed after level 2 as reported by [16].

clustering relevant keypoint SIFT descriptors from the image, we label 3D points with codewords computed by clustering *3D shape context* descriptors [3, 9] computed around the 3D points. In our experiments 3D shape context descriptors were 48-dimensional histograms composed of 3 radial bins and 4×4 angular bins. We used a level-0 3D structure of histograms for capturing the distribution of shape-context codewords. This allows us to make a fair comparison with appearance-based methods. We found a classification accuracy of 41%. This result confirms the superior performance of the occupancy-based 3D structure.

We take note that classifying a query sequence using our SVM-based 3D structure matching scheme is very fast and can be performed in the order of a second on a standard machine. The actual 3D reconstruction of the query video sequence, however, may be more demanding computation-

ally. Even if our current implementation cannot achieve real time reconstruction, recent research [20] has shown that this can be eventually made possible.

5. Conclusions

We have presented a new method for scene categorization from low definition video sequences. As far as we know, our method is one of the first attempts to combine structure (collection of 3D points) with imagery (feature points labeled by codewords) into a single framework for scene categorization. We argue that the underlying 3D structure of the scene can greatly help categorization by capturing the typical distribution of appearance elements in 3D. Our claims are validated by a series of experiments carried out on a challenging dataset of video sequences comprising 5 scene categories. We see this work as a promising starting point toward the goal of designing systems for coherent scene understanding and automatic extraction of the object semantics in the scene.

6. Acknowledgements

We thank Andrey Del Pozo for the hardwork put in collecting the dataset and for insightful suggestions in a preliminary version of this work.

References

[1] http://maps.google.com/help/maps/streetview/.

[2] Three dimensional scene categories video dataset. http://www.eecs.umich.edu/vision/3DSceneDataset.html.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4), 2002.

[4] A. Berg, F. Grabler, and J. Malik. Parsing images of architectural scenes,. In *IEEE 11th International Conference on Computer Vision*, 2007.

[5] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *In Proc. 10th ECCV*, 2008.

[6] M. Brown and D. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *5th International Conference on 3D Imaging and Modelling (3DIM05)*, Ottawa, Canada, 2005.

[7] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2), 2008.

[8] L. Fei-Fei and P. Perona. A Bayesian hierarchy model for learning natural scene categories. *CVPR*, 2005.

[9] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. *ECCV*, 2004.

[10] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, 2005.

[11] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3d object recognition from range images using local feature histograms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 2, 2001.

[12] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *Int. Conf. on Computer Vision*, 2005.

[13] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1), 2008.

[14] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. In *IEEE PAMI*, volume 5, 1999.

[15] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *In Symposium on Geometry Processing*, 2003.

[16] L. Lazebnik, S. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings IEEE Computer Vision and Pattern Recognition*, 2007.

[17] X. Li, I. Guskov, and J. Barhak. Feature-based alignment of range scan data to cad model. In *International Journal of Shape Modeling*, volume 13, pages 1–23, 2007.

[18] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999.

[19] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3), 2007.

[20] D. Nister. Preemptive ransac for live structure and motion estimation. 16, 2005.

[21] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *Image Processing, IEEE Transactions on*, 14(2):169–180, Feb. 2005.

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42:145–175, 2001.

[23] S. Ruiz-Correa, L. Shapiro, and M. Meila. A new signature-based method for efficient 3-d object recognition. In *Proc. In IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[24] S. Saxena, M. Sun, and A. Ng. Make3d: Learning 3-d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[25] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proc. International Conference on Computer Vision*, 2005.

[26] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings*, pages 145–156, 2004.

[27] P. Torr. An assessment of information criteria for motion model selection. In *CVPR*, pages 47–52, Puerto Rico, 1997.

[28] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003.

[29] J. Vogel and B. Schiele. A semantic typicality measure for natural scene categorization. In *DAGM'04 Annual Pattern Recognition Symposium*, Tuebingen, Germany, 2004.